

# **English-to-Dari and Dari-to-English Medical Phrasebook Android Application Software System Documentation**

**by Robert Winkler, Somiya Metu, Dr. Steve LaRocca,  
and Ghulam Hazrat Jahed**

**ARL-TR-5975**

**April 2012**

## **NOTICES**

### **Disclaimers**

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

**DESTRUCTION NOTICE**—For classified documents, follow the procedures in DoD 5220.22-M, National Industrial Security Program Operating Manual, Chapter 5, Section 7, or DoD 5200.1-R, Information Security Program Regulation, C6.7. For unclassified, limited documents, destroy by any method that will prevent disclosure of contents or reconstruction of the document.

# **Army Research Laboratory**

Adelphi, MD 20783-1197

---

**ARL-TR-5975****April 2012**

---

## **English-to-Dari and Dari-to-English Medical Phrasebook Android Application Software System Documentation**

**Robert Winkler, Somiya Metu, Dr. Steve LaRocca,  
and Ghulam Hazrat Jahed  
Computational and Information Sciences Directorate, ARL**

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p><b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>					
1. REPORT DATE (DD-MM-YYYY) April 2012		2. REPORT TYPE Final		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE English-to-Dari and Dari-to-English Medical Phrasebook Android Application Software System Documentation				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Robert Winkler, Somiya Metu, Dr. Steve LaRocca, and Ghulam Hazrat Jahed				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: RDRL-CII-B 2800 Powder Mill Road Adelphi, MD 20783-1197				8. PERFORMING ORGANIZATION REPORT NUMBER  ARL-TR-5975	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT This report describes an Android application for facilitating exchange between English-speaking and Dari-speaking domain experts with the help of mobile devices such as smartphones and tablet computers. The U.S. Army Research Army Laboratory (ARL) has been developing English-to-Dari phrasebooks in a variety of specialized domains (medical, legal, financial, et al.). This application provides bi-directional English-Dari text translation using a dictionary of medical phrases selected and translated from the English-language medical training manual, <i>Fundamental Critical Care Support (FCCS)</i> , published by the Society for Critical Care Medicine (SCCM). The application is database-driven and is domain agnostic and easily adaptable to other specialized domains.					
15. SUBJECT TERMS Soft custom keyboard, phrasebook, Dari reshaper, Android SQLite database					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT  UU	18. NUMBER OF PAGES  60	19a. NAME OF RESPONSIBLE PERSON Robert Winkler
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) (301) 394-1689

---

## Contents

---

<b>List of Figures</b>	<b>iv</b>
<b>Acknowledgments</b>	<b>v</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. Development Environment</b>	<b>2</b>
<b>3. Application Design</b>	<b>3</b>
3.1 The Look and Feel.....	4
3.2 English-Dari Translation .....	5
3.3 Dari-English Translation .....	7
3.4 The Dari Soft Keyboard Design .....	10
3.5 Porting of the Application to Android 3.0 (Honeycomb) .....	12
<b>4. Conclusion and Future Work</b>	<b>14</b>
<b>5. References</b>	<b>15</b>
<b>Appendix A. XML User Interface Layout</b>	<b>17</b>
<b>Appendix B. Source Code</b>	<b>31</b>
<b>Distribution List</b>	<b>51</b>

---

## List of Figures

---

Figure 1. English-Dari medical phrasebook class diagram. ....	3
Figure 2. MainForm screenshot. ....	4
Figure 3. Start of an English-Dari translation. ....	6
Figure 4. Autocomplete suggestions. ....	6
Figure 5. Completed English-Dari translation. ....	6
Figure 6. English to Dari sequence diagram. ....	7
Figure 7. Dari soft keyboard – vertical layout. ....	8
Figure 8. Autocomplete Dari suggestions. ....	8
Figure 9. Completed Dari-English translation. ....	9
Figure 10. Dari to English sequence diagram. ....	10
Figure 11. Dari soft keyboard - horizontal layout (Shift OFF). ....	11
Figure 12. Dari soft keyboard – horizontal layout (Shift ON). ....	11
Figure 13. Dari keyboard with toast. ....	12
Figure 14. English Dari medical phrasebook on Android 3.0 Honeycomb (Soft Dari keyboard displayed). ....	13
Figure 15. English Dari medical phrasebook on Android 3.0 Honeycomb (Soft English keyboard displayed). ....	13

---

## Acknowledgments

---

The content for this Medical Phrasebook Application is an extensive set of technical medical terms extracted from translated texts. To automate the identification of technical terms, the ARL project team made use of the U.S. National Library of Medicine's Unified Medical Language System (UMLS), which includes knowledge sources and advanced tools useful for the development of computer systems that behave as if they understand biomedical language.

We would also like to acknowledge the use of a freely available package called com.ahmadiv.dari.DariGlyphUtils from <http://code.google.com/p/glyph-util/>

Android 2.2 does not display Persian/Arabic letters correctly. This package has been used to connect and reshape Persian/Arabic letters for correct display and legibility.

INTENTIONALLY LEFT BLANK.



---

## 1. Introduction

---

The U.S. Army Research Laboratory's (ARL) Computational and Information Science Directorate (CISD) has been developing translation technologies and translation tools to support security transition operations in Afghanistan. U.S. Soldiers serving on medical Embedded Training Teams working with the medical units of the Afghan National Army have pointed the need to accelerate the transfer of modern medical knowledge to medical schools, physicians and other medical professionals in Afghanistan. High quality translations of medical training texts are urgently needed, but difficult to produce in a country where the educational infrastructure is rebuilding after decades of conflict.

In the course of a project to produce a special trainer's edition of a medical reference manual, the need for a bilingual English-Dari glossary of technical medical terms became very apparent as the project team sought to achieve clarity and consistency across all chapters and appendices of the book. Software designed to help automate the selection of technical terms was pressed into service and a sizeable collection of more than 6,000 English words and phrases peculiar the medical training, each with its Dari equivalent, was the result.

Developed primarily as input for training translation software, the English-Dari medical glossary contains information that is immediately useful for trainers, students, physicians and other medical professionals. While printed paper copies might be one way to share this information, a software version of an English-to-Dari and Dari-to-English Medical Phrasebook offers a number of important advantages, including ease of use and portability. Given the great popularity of cellular phones in Afghanistan and throughout the developing world, a Medical Phrasebook application for the Android operating system presented itself as a very practical means of proliferating and sharing this medical knowledge.

This paper documents the development of an application for a bidirectional English-Dari Medical Phrasebook based on critical care medical terms. Designed for Google's Android platform, it allows data entry and look-up in both languages, making use of either a soft keyboard or a device's hardware according to user preference. The paper first describes the development environment, then moves to the design of the application with a discussion of the rationale behind key design decisions. Finally, we provide the documented source code for the application itself.

---

## 2. Development Environment

---

Google's Android is an open-source software stack intended for mobile devices such as cell phones and tablets. The development environment consists of a Software Development Kit (SDK) written for the Java programming language, and a series of tools integrated with the Eclipse integrated development environment (IDE) for development and debugging. The Android SDK and the associated tools are freely available from Google at <http://developer.android.com>. The version we used for this development was SDK Tools Revision 10 with the Android Development Tools (ADT) Eclipse Plug-in for Eclipse Helios. The target platform was Android 2.2. The Motorola Droid 2 was the phone used for testing.

Additionally, Android 2.2 does not natively support a Persian keyboard or the shaping of Dari characters.\* A package freely available from <http://code.google.com/p/glyph-util/> called com.ahmadiv.dari.DariGlyphUtils was used to reshape the Dari phrases.

---

\* Persian languages are cursive instead of block lettering so the characters must be joined or "shaped".

### 3. Application Design

Figure 1 illustrates the overall class diagram for the application.

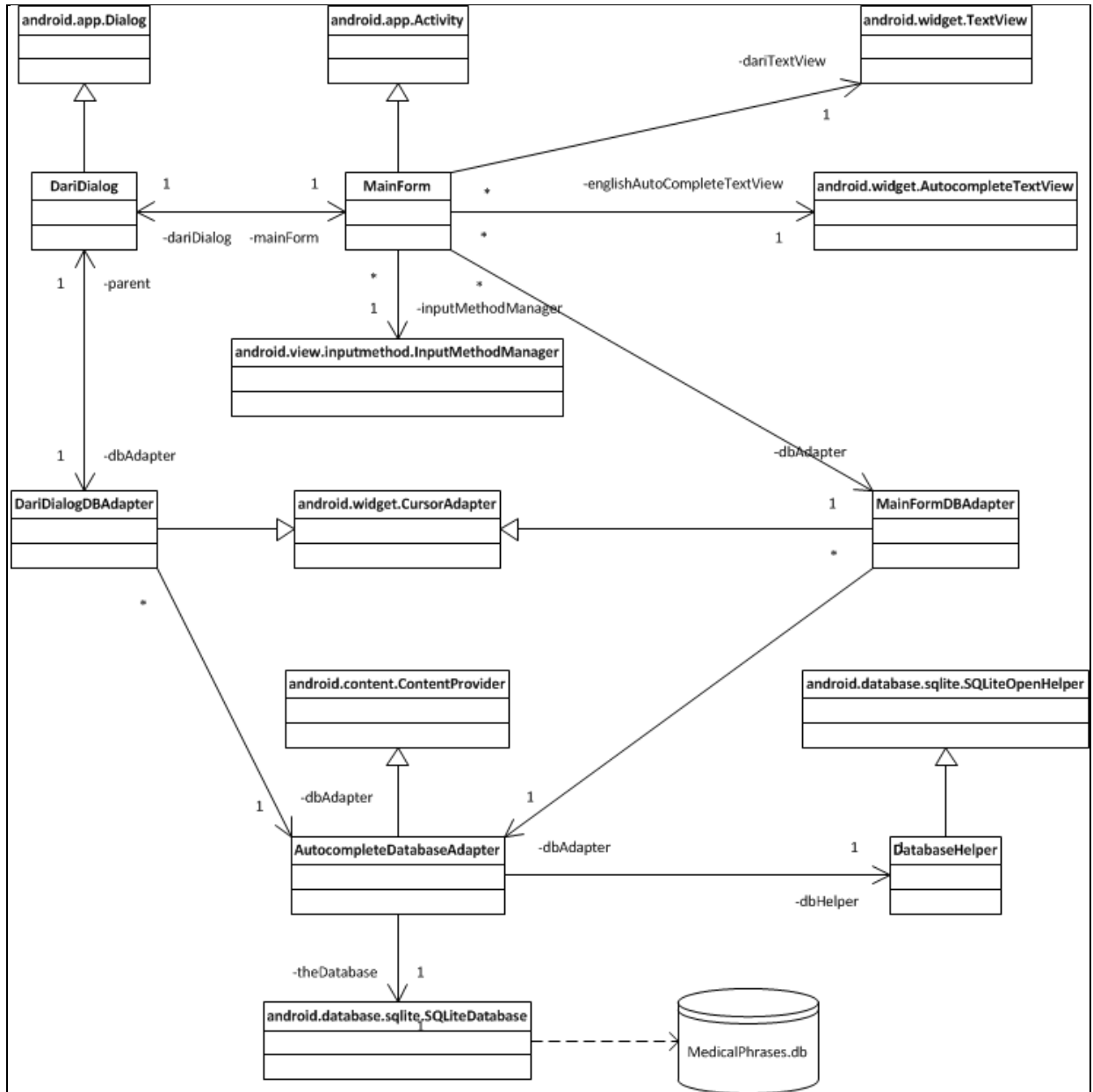


Figure 1. English-Dari medical phrasebook class diagram.

### 3.1 The Look and Feel

MainForm is the Android Activity<sup>†</sup> that presents the user interface (UI) for the English-Dari Medical Phrasebook. MainForm presents the UI through a number of widgets organized in an Android layout. The widgets and layout are specified in an XML file called `main.xml`, which lives in the layout folder under the application's resource folder (called `res`). Our MainForm uses a Linear Layout nested inside a Relative Layout for the entry screen. The UI for our main entry screen annotated with the widgets and layouts is shown in figure 2, and the `main.xml` is listed in the appendix.

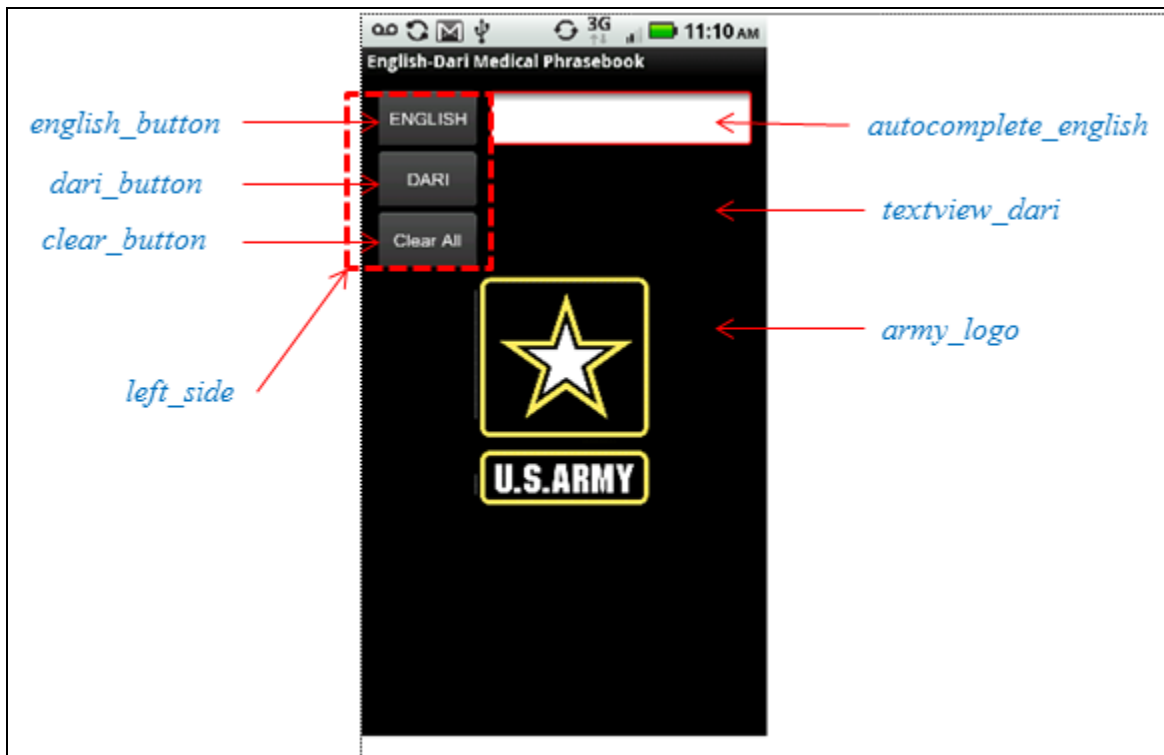


Figure 2. MainForm screenshot.

The MainForm's top-level layout is an Android RelativeLayout. A RelativeLayout allows child widgets and other views to specify their position relative to each or their parent. The buttons are contained within the *left\_side* Android LinearLayout. A LinearLayout aligns the children in either a vertical or horizontal direction. In this case, we set the orientation to vertical and force the child buttons to be set to the same width and height (`android:layout_weight = "1"`). The *english\_button* is used to initiate translations from English to Dari. The *dari\_button* is used to initiate translations from Dari to English. The *clear\_button* clears both the *english\_button* and the *dari\_button* to prepare for a new translation. The *autocomplete\_english* widget is an Android

---

<sup>†</sup> "An Android Activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with `setContentView(View)`." (<http://developer.android.com/reference/android/app/Activity.html>) The application's main activity is typically the entry screen.

AutoCompleteTextView, which shows completion suggestions from the list of English medical phrases after the user has typed in at least three characters (`android:completionThreshold = "3"`). It is used to specify the input phrase for an English-Dari translation and display the result of a Dari-English translation. The *textview\_dari* widget is an Android TextView that is set to be uneditable (`android:editable = "false"`). It is used to display the result of an English-Dari translation and to capture the input from the custom Dari soft keyboard (described later) used to specify the input for a Dari-English translation.

### 3.2 English-Dari Translation

An English-Dari translation is initiated by either pressing the *english\_button* or clicking in the *autocomplete\_english* text view. Either action will bring up the Android's native soft keyboard as illustrated in figure 3. Once the user has entered at least three characters that match the beginning of an English medical phrase in the database, a list of potential matching phrases drops down beneath the *autocomplete\_english* widget as shown in figure 4. Once at least three characters have been entered into the *autocomplete\_english* AutoCompleteTextView, it calls `runQueryOnBackgroundThread` on a `MainFormDBAdapter` with the characters entered so far. The `MainFormDBAdapter` is an Android CursorAdapter used to bind the ListView associated with the drop-down suggestion list for *autocomplete\_english* with a database cursor. The `MainFormDBAdapter` then calls an `AutocompleteDatabaseAdapter` with the partial English phrase. The `AutocompleteDatabaseAdapter` is an Android ContentProvider, which in this case encapsulates the SQLite database containing the English and Dari medical phrases. The `AutocompleteDatabaseAdapter` issues an SQL query of the form "SELECT english, dari FROM phrase where english LIKE partialPhrase%". If the query returns a non-empty result set, the `AutocompleteDatabaseAdapter` calls the Android Activity (in this case the `MainForm`) to `startManagingCursor`, which results in a drop-down list of suggestions appearing below the *autocomplete\_english* AutoCompleteTextView. Once the user has clicked on a suggestion, the `onItemClick` method of the *autocomplete\_english* is invoked. The `onItemClick` method then asks for the English phrase from the database cursor and sets its text to the corresponding string. It then retrieves the corresponding Dari phrase from the same cursor, calls the `DariGlyphUtils` to reshape the Dari string, and then uses that to set the text on the *dari\_textview* widget. Once the translation has been retrieved, the drop-down disappears, the English phrase is captured in the *autocomplete\_english* text view, and the Dari translation is captured in the *textview\_dari* text view as shown in figure 5. This comprises one iteration of an English-to-Dari translation. The UML Sequence Diagram for this use case is shown in figure 6.

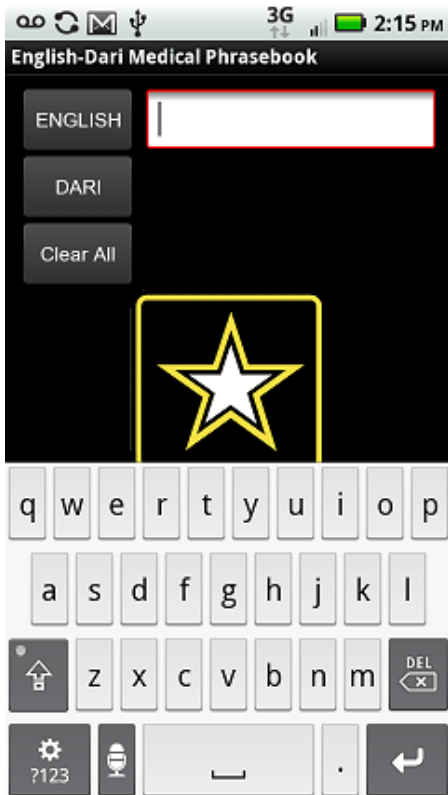


Figure 3. Start of an English-Dari translation.

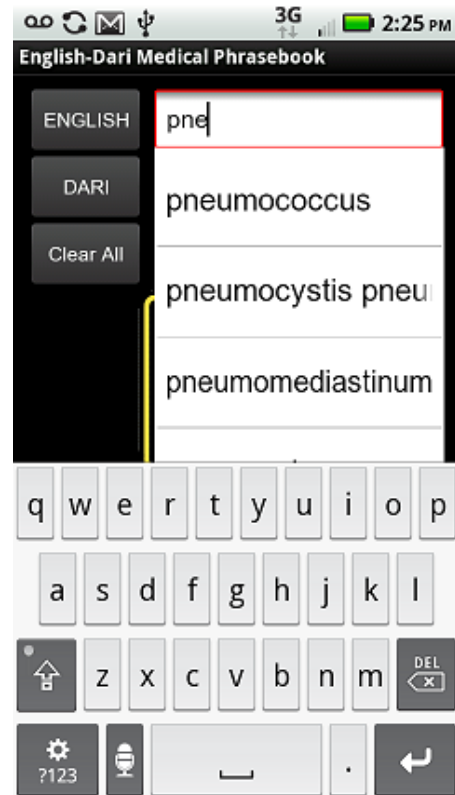


Figure 4. Autocomplete suggestions.

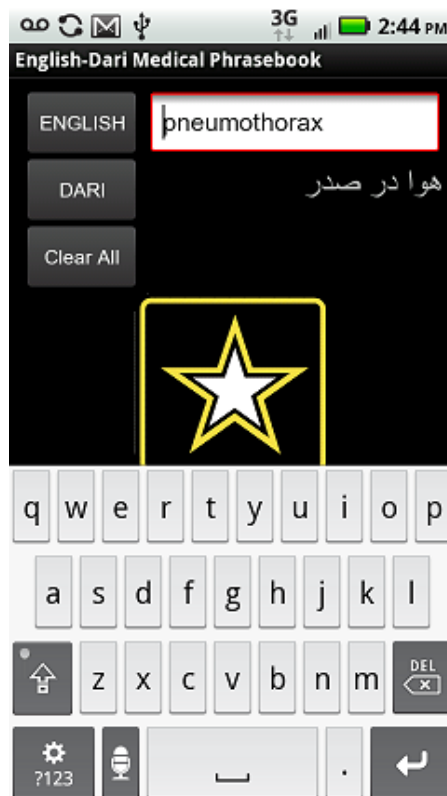


Figure 5. Completed English-Dari translation.

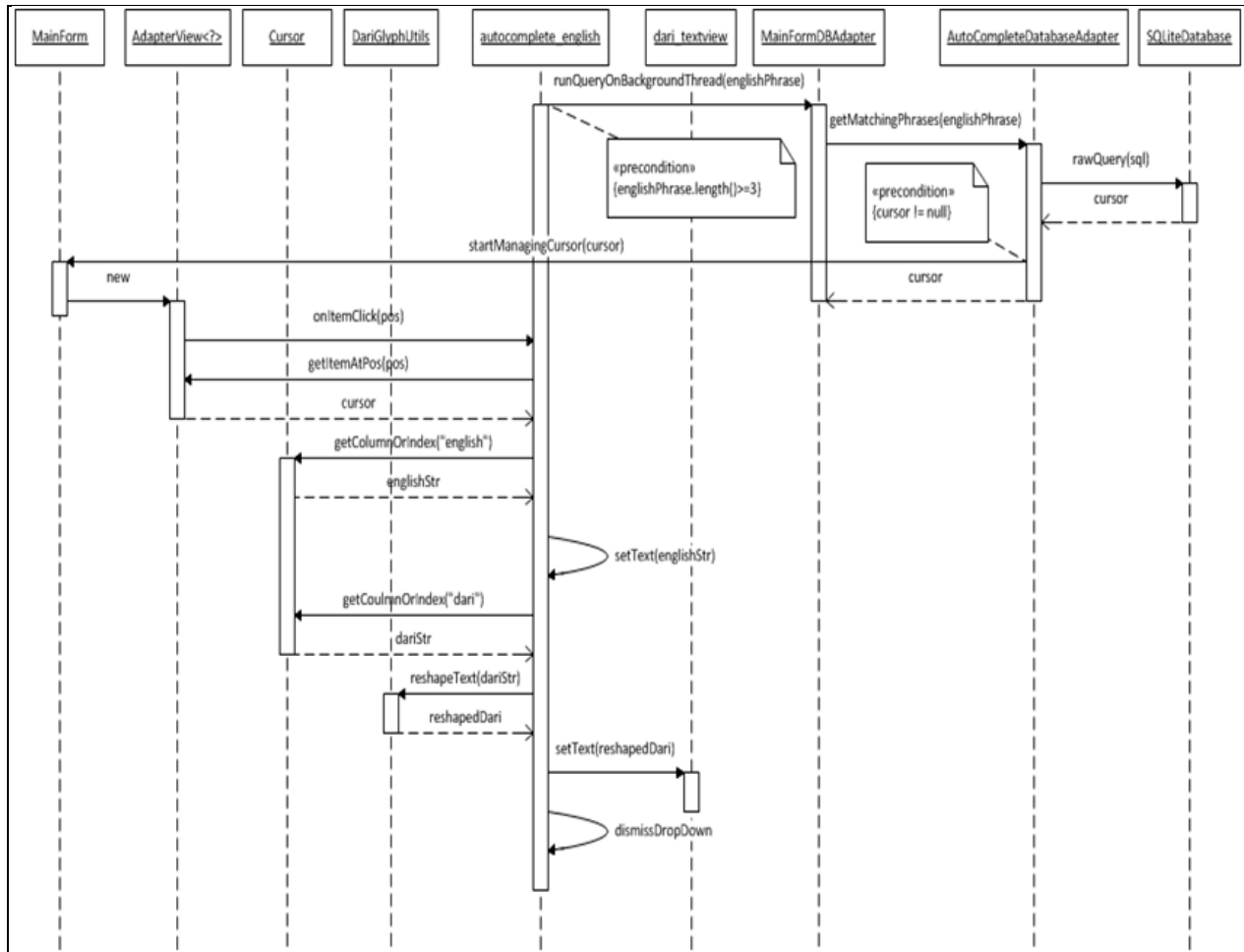


Figure 6. English to Dari sequence diagram.

### 3.3 Dari-English Translation

Translation of medical phrases from Dari into English is initiated by clicking the *dari\_button*. Dari-English translation is complicated by the lack of native support for Persian languages in Android 2.2. Third-party Persian soft keyboards are available, but due to platform limitations the user must manually go into the device's settings to switch between languages. Since manually switching keyboards during a single translation is an unacceptable burden on the user, this required the creation of a custom soft keyboard for the Dari character set. The design of this keyboard is discussed in a later section.

Clicking on the *dari\_button* will bring up the custom soft Dari keyboard as illustrated in figure 7. Once the user has entered at least three characters that match the beginning of a Dari medical phrase in the database, a list of potential matching phrases drops down beneath the *autocomplete\_dari* widget in the Dari Keyboard as shown in figure 8. Once at least three characters have been entered into the *autocomplete\_dari* *AutoCompleteTextView*, the database is queried to display potential matching phrases in a similar manner to that described for the

English-Dari translation. Once a Dari phrase has been selected the database is queried (again in a similar manner as that described for the English-Dari translation) to find the matching English phrase. Once the translation has been retrieved, the Dari soft keyboard and drop-down disappear, the Dari phrase is captured in the *dari\_textview* text view, and the English translation is captured in the *autocomplete\_english* text view, as shown in figure 9. This comprises one iteration of a Dari-to-English translation. The UML Sequence Diagram for this use case is shown in figure 10.

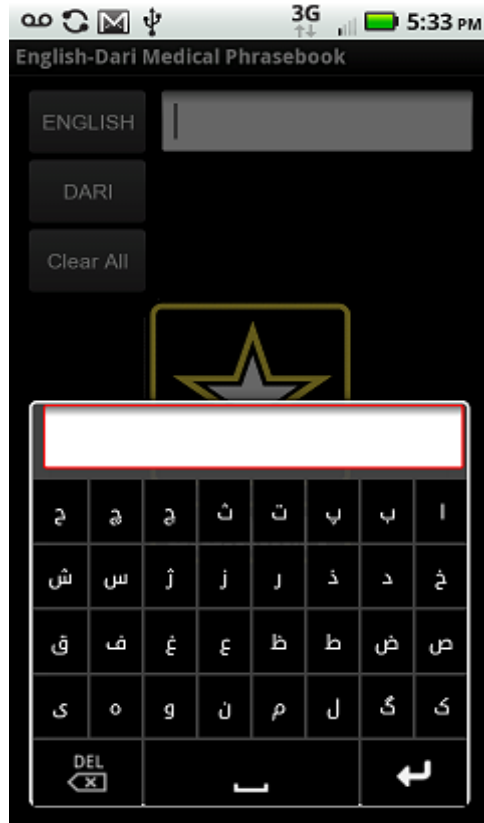


Figure 7. Dari soft keyboard – vertical layout. Figure 8. Autocomplete Dari suggestions.





Figure 9. Completed Dari-English translation.

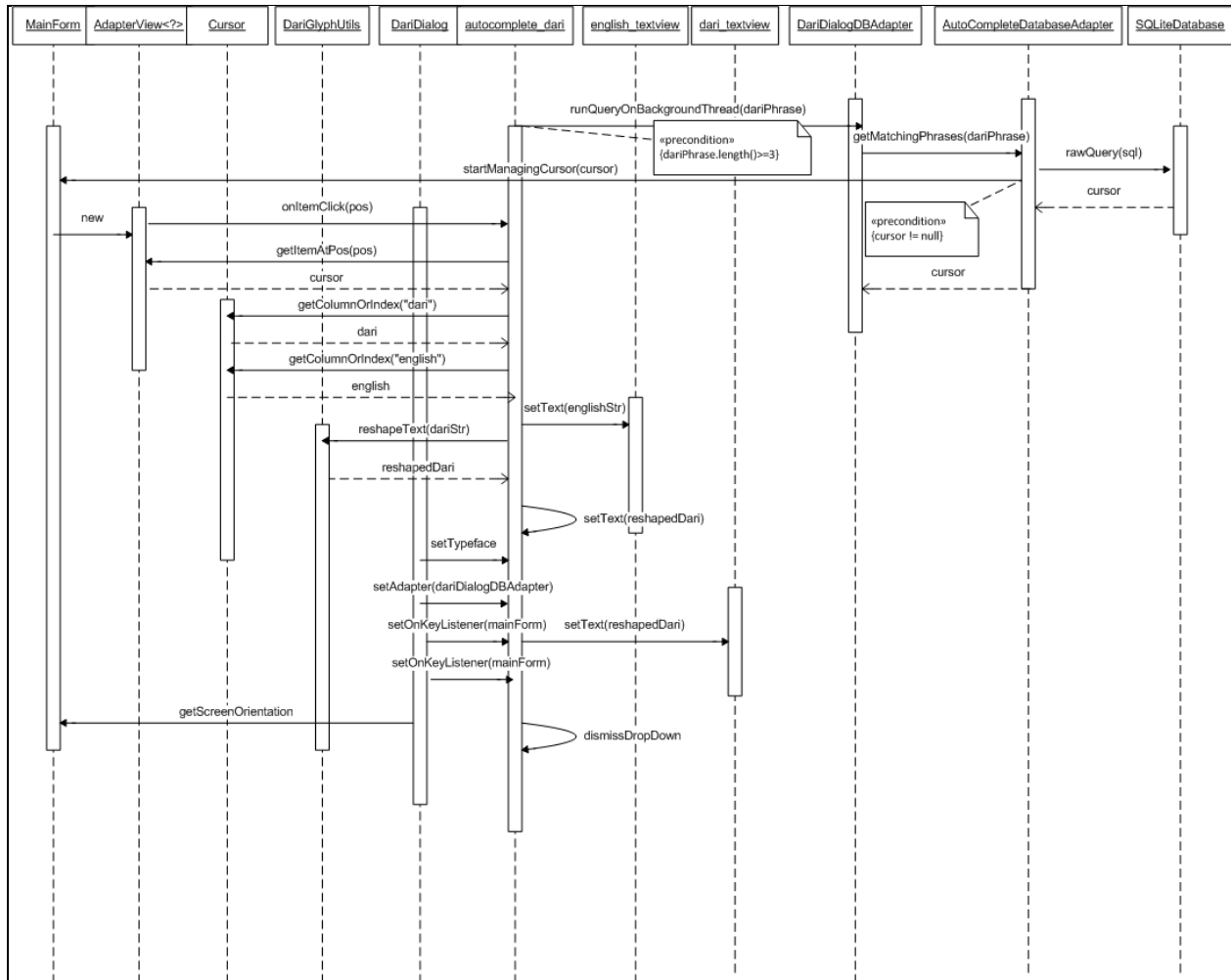


Figure 10. Dari to English sequence diagram.

### 3.4 The Dari Soft Keyboard Design

If a hard keyboard exists on the device, we would like the Dari soft keyboard to have a natural, intuitive mapping to the hard keyboard. Complicating the design of the Dari soft keyboard was the fact that 32 characters are used instead of the 26 used in English. We implemented two different layouts for the Dari soft keyboard using the following assumptions. If the device is held in a vertical orientation, then any hard keyboard associated with the device will typically not be available, and it is unnecessary to accommodate a natural, intuitive mapping to the hard keyboard layout. On the other hand, if the device is held in a horizontal orientation, then the hard keyboard is likely to be accessible, and a different layout that tries to mimic the hard keyboard layout in a natural and intuitive way should be preferred. Figure 7 shows the Dari soft keyboard in the vertical orientation. Figures 11 and 12 show the layout in the horizontal orientation. Since Dari uses six more characters than English, the shift key was employed to introduce the additional characters. The positioning and mapping of the characters to keys was copied from existing Dari keyboard layouts designed for desktop computers. When the shift key is off, the top

row on the Dari soft keyboard maps to the top row of letters on the QWERTY keyboard. The second row on the Dari keyboard maps to the middle row of letters row on the QWERTY keyboard. And the third row on the Dari keyboard maps to the bottom row on the QWERTY keyboard. When the shift key is on, the Dari characters displayed are mapped to the ‘W’, ‘T’, ‘I’, ‘S’, ‘G’, and ‘Z’ keys, mimicking the layout on Dari keyboards designed for desktop PCs. Since it is often difficult to tell exactly which key is being pressed on soft keyboards, one additional feature we wanted to mimic from the native soft keyboard was to temporarily pop up a displaced and enlarged view of the key being pressed. We used an Android Toast for this purpose. Figure 13 illustrates this in action of the Dari soft keyboard.

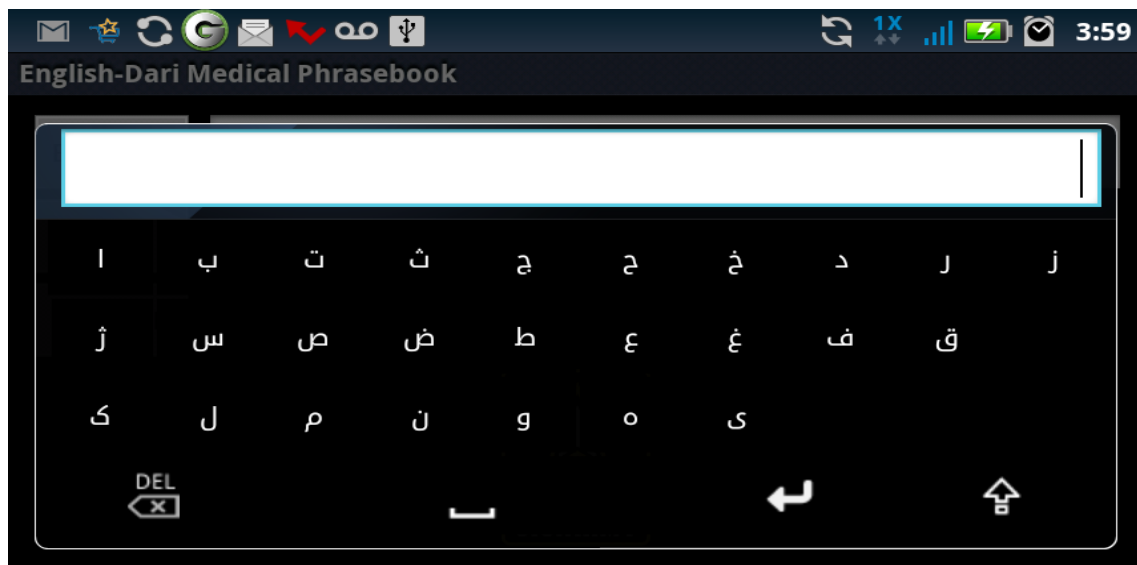


Figure 11. Dari soft keyboard - horizontal layout (Shift OFF).

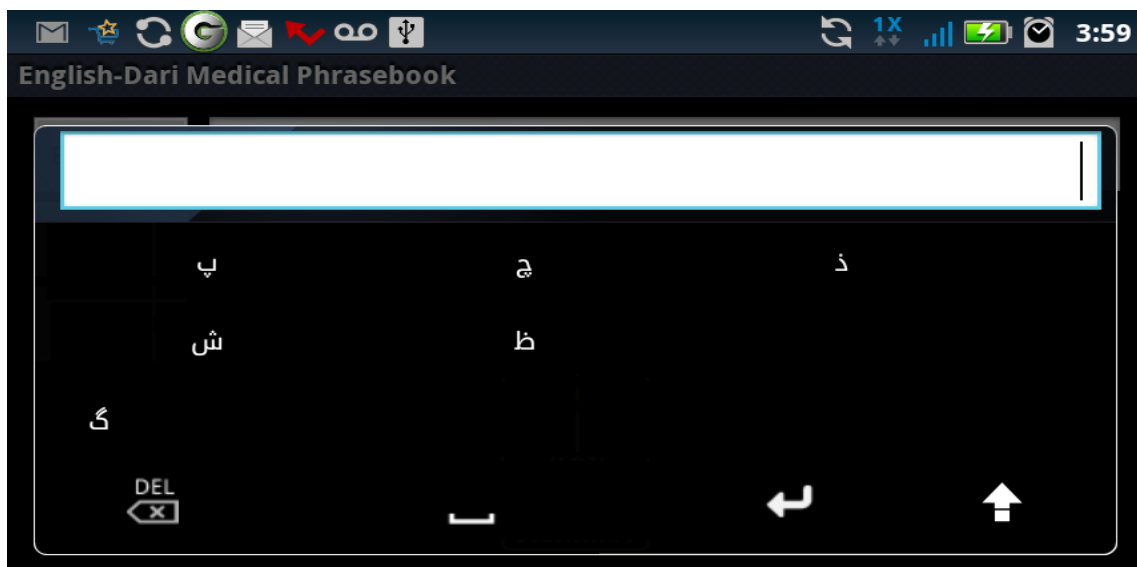


Figure 12. Dari soft keyboard – horizontal layout (Shift ON).

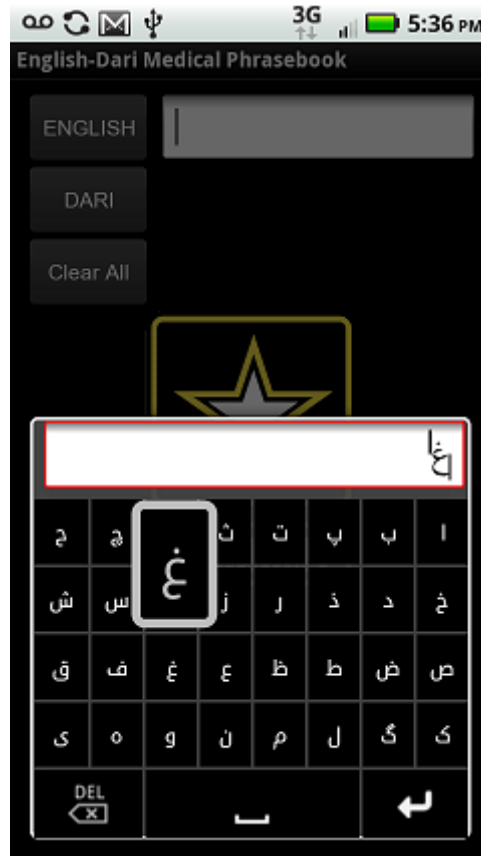


Figure 13. Dari keyboard with toast.

### 3.5 Porting of the Application to Android 3.0 (Honeycomb)

The English Dari phrasebook application has been ported to Android 3.0 Honeycomb. This version of the Android operating system supports larger screens like those found on tablets. With Honeycomb, the tablets do not need physical buttons/hard keyboard; the soft keyboard buttons appear on the bottom of the screen regardless of which way the device is oriented. This eliminates the need of intuitive mapping of Dari soft keyboard to the hard keyboard. Figure 14 shows the look and feel of Soft Dari keyboard when displayed on a tablet. Figure 15 shows the native soft English keyboard.

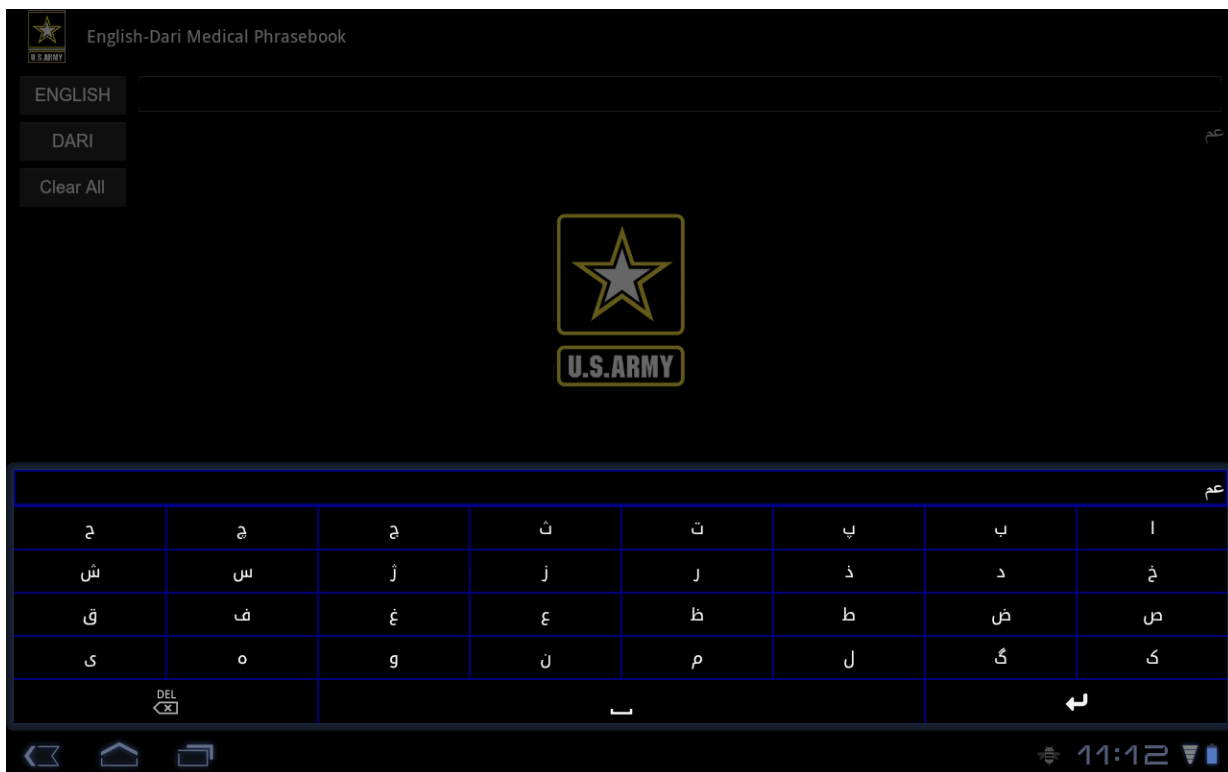


Figure 14. English Dari medical phrasebook on Android 3.0 Honeycomb (Soft Dari keyboard displayed).

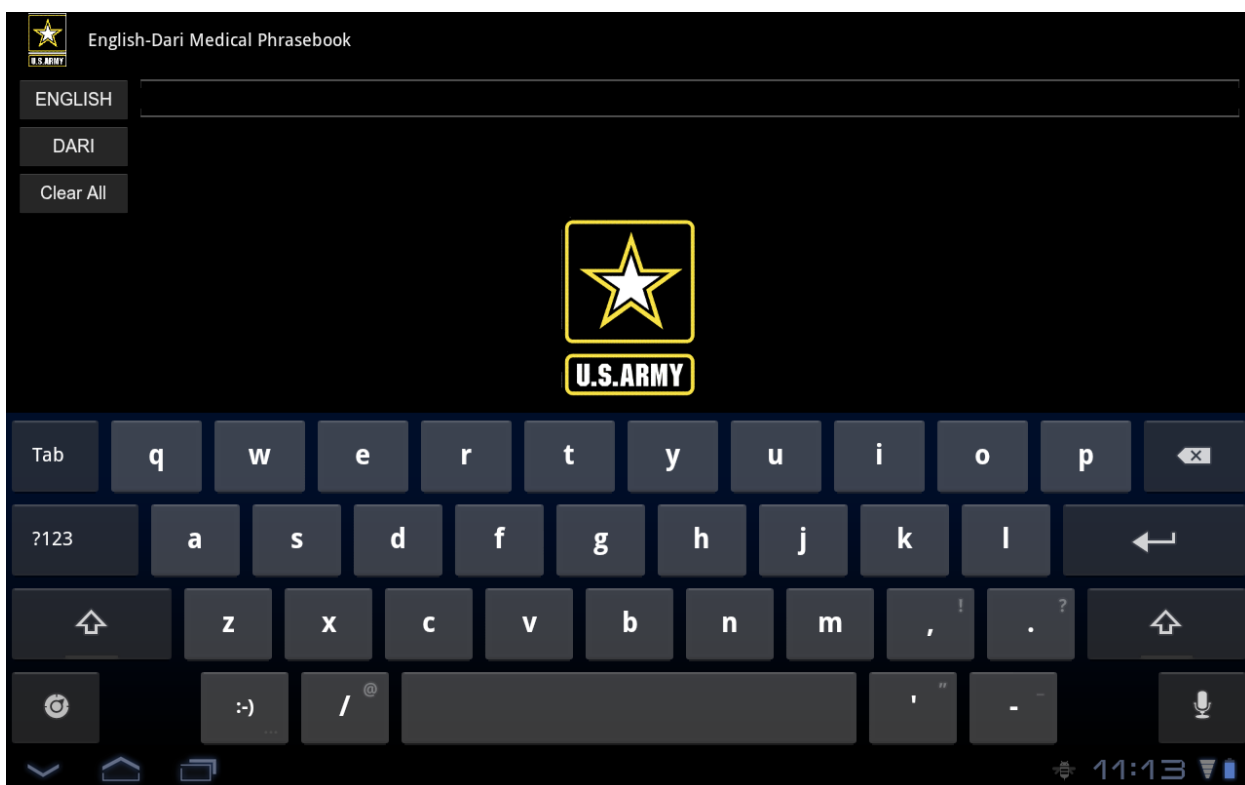


Figure 15. English Dari medical phrasebook on Android 3.0 Honeycomb (Soft English keyboard displayed).

---

## 4. Conclusion and Future Work

---

We have presented the design and implementation of an Android application for the translation of a vetted, manually translated suite of some 6,000 English medical phrases from *Fundamental Critical Care Support (FCCS)*, published by the Society for Critical Care Medicine (SCCM), to Dari and from Dari back into English. The application is database-driven, meaning that new domains can be added or substituted by merely changing or replacing the database. Additionally, we are in the process of collecting a speech database to augment the translation by playing a recording of the translated phrase when it is displayed.

---

## 5. References

---

1. Morgan, John J. *Human in the Loop Machine Translation of Medical Terminology*; ARL-MR-0743; U.S. Army Research Laboratory: Adelphi, MD 20783, April 2010.

INTENTIONALLY LEFT BLANK.



---

## **Appendix A. XML User Interface Layout**

---

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal"
    android:padding="10dp" >
    <LinearLayout android:id="@+id/left_side"
        android:orientation="vertical"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <Button android:id="@+id/english_button"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:layout_weight="1"
            android:onClick="ShowEnglishKeyboard"
            android:text="ENGLISH"
            android:layout_alignParentLeft="true" />
        <Button android:id="@+id/dari_button"
            android:layout_width="fill_parent"
            android:layout_weight="1"
            android:layout_height="fill_parent"
            android:text="DARI"
            android:layout_alignLeft="@id/english_button"
            android:onClick="ShowDariKeyboard"
            android:layout_below="@id/english_button" />
        <Button android:id="@+id/clear_button"
            android:layout_width="fill_parent"
            android:layout_weight="1"
            android:onClick="ClearAll"
            android:layout_height="fill_parent"
            android:layout_below="@id/dari_button"
            android:text="Clear All" />
    </LinearLayout>
    <AutoCompleteTextView android:id="@+id/autocomplete_english"
        android:completionThreshold="3"
        android:layout_toRightOf="@id/left_side"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp" />
    <TextView android:id="@+id/textview_dari"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/autocomplete_english"
        android:editable="false"
        android:layout_alignLeft="@id/autocomplete_english"
        android:gravity="right"
        android:focusable="true"
        android:textSize="24dp"/>
    <ImageView android:id="@+id/army_logo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:src="@drawable/armypic5"
        android:layout_below="@id/left_side"
        android:scaleType="centerInside"
        android:focusable="true"/>
</RelativeLayout>

```

## MainForm Activity Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/help_layout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:padding="10dp" >
    <TextView android:id="@+id/help_line1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:editable="false"
        android:text="@string/organization"
        android:textSize="18dp"/>
    <TextView android:id="@+id/help_line2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:editable="false"
        android:text="@string/poc"
        android:textSize="18dp"/>
    <TextView android:id="@+id/help_line3"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:editable="false"
        android:text="@string/email"
        android:autoLink="email"
        android:textSize="18dp"/>
</LinearLayout>
```

## Help Layout

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:stretchColumns="*">
    <TableRow android:id="@+id/r0" android:layout_width="fill_parent"
        android:layout_height="fill_parent" >
        <AutoCompleteTextView android:id="@+id/autotextview_dari"
            android:layout_width="fill_parent" android:layout_height="wrap_content"
            android:layout_marginLeft="5dp" android:gravity="right"
            android:editable="false" android:textSize="24dp" android:layout_span="10" />
    </TableRow>
    <TableRow android:id="@+id/r1" android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dp">
            <Button android:id="@+id/keypad_1" android:layout_width="match_parent"
                android:layout_height="match_parent" android:layout_marginTop="2dp"
                android:onClick="DariKeyboardClick"
                android:background="@drawable/my_btn"
                android:textColor="@android:color/white" android:text="\u0627" />
        </LinearLayout>
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dp">
            <Button android:id="@+id/keypad_2" android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginTop="2dp"
                android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
                android:background="@drawable/my_btn"
                android:textColor="@android:color/white"
                android:text="\u0628" />
        </LinearLayout>
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dp">
            <Button android:id="@+id/keypad_3" android:layout_width="match_parent"
                android:layout_height="match_parent" android:layout_marginTop="2dp"
                android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
                android:background="@drawable/my_btn"
                android:textColor="@android:color/white"
                android:padding="0dp" android:text="\u062A" />
        </LinearLayout>
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dp">
            <Button android:id="@+id/keypad_4" android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginTop="2dp"
                android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
                android:background="@drawable/my_btn"
                android:textColor="@android:color/white"
                android:text="\u062B" />
        </LinearLayout>
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dp">
            <Button android:id="@+id/keypad_5" android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginTop="2dp"
                android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
                android:background="@drawable/my_btn"
                android:textColor="@android:color/white"
                android:text="\u062C" />
        </LinearLayout>
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dp">
            <Button android:id="@+id/keypad_6" android:layout_width="match_parent"
                android:layout_height="match_parent" android:layout_marginTop="2dp"
                android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
                android:background="@drawable/my_btn"
                android:textColor="@android:color/white"
                android:padding="0dp" android:text="\u062D" />
        </LinearLayout>
    </TableRow>
</TableLayout>
```

## Dari Keyboard Layout

```
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_7" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@drawable/my_btn"
        android:textColor="@android:color/white"
        android:text="\u062E" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_8" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@drawable/my_btn"
        android:textColor="@android:color/white"
        android:text="\u062F" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_9" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:onClick="DariKeyboardClick"
        android:background="@drawable/my_btn"
        android:textColor="@android:color/white" android:text="\u0631" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_10" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@drawable/my_btn"
        android:textColor="@android:color/white"
        android:text="\u0632" />
</LinearLayout>
</TableRow>
<TableRow android:id="@+id/r2" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_11" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
            android:background="@drawable/my_btn"
            android:textColor="@android:color/white"
            android:text="\u0698" />
        </LinearLayout>
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dip">
            <Button android:id="@+id/keypad_12" android:layout_width="match_parent"
                android:layout_height="match_parent" android:layout_marginTop="2dp"
                android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
                android:background="@drawable/my_btn"
                android:textColor="@android:color/white"
                android:text="\u0633" />
            </LinearLayout>
    </TableRow>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_13" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@drawable/my_btn"
        android:textColor="@android:color/white"
        android:text="\u0635" />
</LinearLayout>
```

## Dari Keyboard Layout Cont.

```
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_14" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@drawable/my_btn"
        android:textColor="@android:color/white"
        android:text="\u00636" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_15" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@drawable/my_btn"
        android:textColor="@android:color/white"
        android:text="\u00637" />
</LinearLayout>

<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_16" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@drawable/my_btn"
        android:textColor="@android:color/white"
        android:text="\u00639" />
</LinearLayout>

<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_17" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:onClick="DariKeyboardClick"
        android:background="@drawable/my_btn"
        android:textColor="@android:color/white" android:text="\u0063A" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_18" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:onClick="DariKeyboardClick"
        android:background="@drawable/my_btn"
        android:textColor="@android:color/white" android:text="\u00641" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_19" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@drawable/my_btn"
        android:textColor="@android:color/white"
        android:text="\u00642" />
</LinearLayout>
```

## Dari Keyboard Layout Cont.

```
<TableRow android:id="@+id/r3" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_20" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:onClick="DariKeyboardClick"
            android:background="@drawable/my_btn"
            android:textColor="@android:color/white" android:text="\u006A9" />
    </LinearLayout>
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_21" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
            android:background="@drawable/my_btn"
            android:textColor="@android:color/white"
            android:text="\u00644" />
    </LinearLayout>
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_22" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
            android:background="@drawable/my_btn"
            android:textColor="@android:color/white"
            android:text="\u00645" />
    </LinearLayout>
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_23" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
            android:background="@drawable/my_btn"
            android:textColor="@android:color/white"
            android:text="\u00646" />
    </LinearLayout>
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_24" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
            android:background="@drawable/my_btn"
            android:textColor="@android:color/white"
            android:text="\u00648" />
    </LinearLayout>
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_25" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
            android:background="@drawable/my_btn"
            android:textColor="@android:color/white"
            android:text="\u00647" />
    </LinearLayout>
```

## Dari Keyboard Layout Cont.

```
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dip">
            <Button android:id="@+id/keypad_26" android:layout_width="match_parent"
                android:layout_height="match_parent" android:layout_marginTop="2dp"
                android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
                android:background="@drawable/my_btn"
                android:textColor="@android:color/white"
                android:text="\u06CC" />
        </LinearLayout>
    </TableRow>

    <TableRow android:id="@+id/r5" android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <ImageButton android:id="@+id/keypad_Del"
            android:layout_width="36dp" android:layout_height="45dip"
            android:onClick="DariKeyboardDeleteClick" android:layout_marginTop="2dp"
            android:src="@drawable/sym_keyboard_delete"
            android:background="@color/candidate_normal"
            android:layout_span="2" />
        <ImageButton android:id="@+id/keypad_Space"
            android:layout_width="36dp" android:layout_height="45dip"
            android:layout_marginTop="2dp" android:layout_marginLeft="2dp"
            android:onClick="DariKeyboardSpaceClick"
            android:src="@drawable/sym_keyboard_space"
            android:background="@color/candidate_normal" android:layout_span="4" />
        <ImageButton android:id="@+id/keypad_Enter"
            android:layout_width="36dp" android:layout_height="45dip"
            android:layout_marginTop="2dp" android:layout_marginLeft="2dp"
            android:onClick="DariKeyboardDoneClick"
            android:src="@drawable/sym_keyboard_return"
            android:background="@color/candidate_normal" android:layout_span="2" />
        <ImageButton android:id="@+id/keypad_Shift"
            android:layout_width="36dp" android:layout_height="45dip"
            android:layout_marginTop="2dp" android:layout_marginLeft="2dp"
            android:onClick="DariKeyboardShiftClick"
            android:src="@drawable/sym_keyboard_shift"
            android:background="@color/candidate_normal" android:layout_span="2" />
    </TableRow>
</TableLayout>
```



## Dari Keyboard Layout Cont.

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:stretchColumns="*">
    <TableRow android:id="@+id/r0" android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <AutoCompleteTextView android:id="@+id/autocompleteview_dari"
            android:layout_width="fill_parent" android:layout_height="wrap_content"
            android:layout_marginLeft="5dp" android:gravity="right"
            android:editable="false" android:textSize="24dp" android:layout_span="8" />
    </TableRow>
    <TableRow android:id="@+id/r1" android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dp">
            <Button android:id="@+id/keypad_8" android:layout_width="match_parent"
                android:layout_height="match_parent" android:layout_marginTop="2dp"
                android:onClick="DariKeyboardClick"
                android:background="@color/candidate_normal"
                android:textColor="@android:color/white" android:text="\u062D" />
        </LinearLayout>
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dp">
            <Button android:id="@+id/keypad_7" android:layout_width="match_parent"
                android:layout_height="match_parent" android:layout_marginTop="2dp"
                android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
                android:background="@color/candidate_normal"
                android:textColor="@android:color/white"
                android:text="\u0686" />
        </LinearLayout>
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dp">
            <Button android:id="@+id/keypad_6" android:layout_width="match_parent"
                android:layout_height="match_parent" android:layout_marginTop="2dp"
                android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
                android:background="@color/candidate_normal"
                android:textColor="@android:color/white"
                android:padding="0dp" android:text="\u062C" />
        </LinearLayout>
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dp">
            <Button android:id="@+id/keypad_5" android:layout_width="match_parent"
                android:layout_height="match_parent" android:layout_marginTop="2dp"
                android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
                android:background="@color/candidate_normal"
                android:textColor="@android:color/white"
                android:text="\u062B" />
        </LinearLayout>
        <LinearLayout android:layout_width="29dp"
            android:layout_height="45dp">
            <Button android:id="@+id/keypad_4" android:layout_width="match_parent"
                android:layout_height="match_parent" android:layout_marginTop="2dp"
                android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
                android:background="@color/candidate_normal"
                android:textColor="@android:color/white"
                android:text="\u062A" />
        </LinearLayout>
    </TableRow>
</TableLayout>
```

## Dari Special Keyboard Layout

```
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_3" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:padding="0dp" android:text="\u067E" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_2" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u0628" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_1" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u0627" />
</LinearLayout>
</TableRow>

<TableRow android:id="@+id/r2" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_16" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:onClick="DariKeyboardClick"
            android:background="@color/candidate_normal"
            android:textColor="@android:color/white" android:text="\u0634" />
    </LinearLayout>
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_15" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
            android:background="@color/candidate_normal"
            android:textColor="@android:color/white"
            android:text="\u0633" />
    </LinearLayout>
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_14" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
            android:background="@color/candidate_normal"
            android:textColor="@android:color/white"
            android:text="\u0698" />
    </LinearLayout>
</TableRow>
```

## Dari Special Keyboard Layout Cont.

```
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_13" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u0632" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_12" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u0631" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_11" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u0630" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_10" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u062F" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_9" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u062E" />
</LinearLayout>
</TableRow>

<TableRow android:id="@+id/r3" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_24" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:onClick="DariKeyboardClick"
            android:background="@color/candidate_normal"
            android:textColor="@android:color/white" android:text="\u0642" />
    </LinearLayout>
```

## Dari Special Keyboard Layout Cont.

```
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_23" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u0641" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_22" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u063A" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_21" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u0639" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_20" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u0638" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_19" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u0637" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_18" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u0636" />
</LinearLayout>
```

## Dari Special Keyboard Layout Cont.

```
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_17" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u0635" />
</LinearLayout>
</TableRow>
<TableRow android:id="@+id/r4" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_32" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:onClick="DariKeyboardClick"
            android:background="@color/candidate_normal"
            android:textColor="@android:color/white" android:text="\u06CC" />
    </LinearLayout>
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_31" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
            android:background="@color/candidate_normal"
            android:textColor="@android:color/white"
            android:text="\u0647" />
    </LinearLayout>
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_30" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
            android:background="@color/candidate_normal"
            android:textColor="@android:color/white"
            android:text="\u0648" />
    </LinearLayout>
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_29" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
            android:background="@color/candidate_normal"
            android:textColor="@android:color/white"
            android:text="\u0646" />
    </LinearLayout>
    <LinearLayout android:layout_width="29dp"
        android:layout_height="45dip">
        <Button android:id="@+id/keypad_28" android:layout_width="match_parent"
            android:layout_height="match_parent" android:layout_marginTop="2dp"
            android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
            android:background="@color/candidate_normal"
            android:textColor="@android:color/white"
            android:text="\u0645" />
    </LinearLayout>
</TableRow>
```

## Dari Special Keyboard Layout Cont.

```
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_27" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u0644" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_26" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u06AF" />
</LinearLayout>
<LinearLayout android:layout_width="29dp"
    android:layout_height="45dip">
    <Button android:id="@+id/keypad_25" android:layout_width="match_parent"
        android:layout_height="match_parent" android:layout_marginTop="2dp"
        android:layout_marginLeft="2dp" android:onClick="DariKeyboardClick"
        android:background="@color/candidate_normal"
        android:textColor="@android:color/white"
        android:text="\u06A9" />
</LinearLayout>
</TableRow>
<TableRow android:id="@+id/r5" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <ImageButton android:id="@+id/keypad_Del"
        android:layout_width="36dp" android:layout_height="45dip"
        android:onClick="DariKeyboardDeleteClick" android:layout_marginTop="2dp"
        android:src="@drawable/sym_keyboard_delete"
        android:background="@color/candidate_normal"
        android:layout_span="2" />
    <ImageButton android:id="@+id/keypad_Space"
        android:layout_width="36dp" android:layout_height="45dip"
        android:layout_marginTop="2dp" android:layout_marginLeft="2dp"
        android:onClick="DariKeyboardSpaceClick"
        android:src="@drawable/sym_keyboard_space"
        android:background="@color/candidate_normal" android:layout_span="4" />
    <ImageButton android:id="@+id/keypad_Enter"
        android:layout_width="36dp" android:layout_height="45dip"
        android:layout_marginTop="2dp" android:layout_marginLeft="2dp"
        android:onClick="DariKeyboardDoneClick"
        android:src="@drawable/sym_keyboard_return"
        android:background="@color/candidate_normal" android:layout_span="2" />
</TableRow>
</TableLayout>
```

---

## Appendix B. Source Code

---

### MainForm.java

```
package mil.army.arl;
import java.util.HashMap;
import java.util.Map;
import com.ahmadiv.dari.DariGlyphUtils;
import android.graphics.Typeface;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.res.Configuration;
import android.database.Cursor;
import android.os.Bundle;
import android.view.Display;
import android.view.Gravity;
import android.view.KeyEvent;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.view.inputmethod.InputMethodManager;
import android.widget.AdapterView;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.CursorAdapter;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ImageButton;

/** This is the main class of the application which extends the Activity class. This class is responsible for providing a screen for
user interaction. It creates a window for drawing the User Interface of the application.
*/
public class MainForm extends Activity implements OnClickListener{

    // Fields
    private Typeface arial;
    public Typeface getArial()
    { // give access to the DariDialogDBAdapter as well
        return arial;
    }
    private TextView dariTextView;
    public TextView getDariTextView()
    { // give access to the DariDialogDBAdapter as well
        return dariTextView;
    }

    private AutoCompleteTextView englishAutoCompleteTextView;
    public AutoCompleteTextView getEnglishAutoCompleteTextView()
    { // give access to the DariDialogDBAdapter as well
        return englishAutoCompleteTextView;
    }
    private InputMethodManager inputMethodManager;
    private DariDialog dariDialog;
```

```

private static boolean dariDialogOpen = false;
private AutocompleteDatabaseAdapter databaseAdapter;
private boolean shiftLocked=false;
// DariDialog stuff - callbacks specified in dari_keyboard.xml and dari_special_keys.xml
private Map<Integer, String> dariKeys = new HashMap<Integer, String>(32, 1.0f);
private void LoadDariKeyMap()
{
    int offset=KeyEvent.getMaxKeyCode();
    dariKeys.put(KeyEvent.KEYCODE_Q, "\u0627");
    dariKeys.put(KeyEvent.KEYCODE_W, "\u0628");
    dariKeys.put(KeyEvent.KEYCODE_W+offset, "\u067E");
    dariKeys.put(KeyEvent.KEYCODE_E, "\u062A");
    dariKeys.put(KeyEvent.KEYCODE_R, "\u062B");
    dariKeys.put(KeyEvent.KEYCODE_T, "\u062C");
    dariKeys.put(KeyEvent.KEYCODE_T+offset, "\u0686");
    dariKeys.put(KeyEvent.KEYCODE_Y, "\u062D");
    dariKeys.put(KeyEvent.KEYCODE_U, "\u062E");
    dariKeys.put(KeyEvent.KEYCODE_I, "\u062F");
    dariKeys.put(KeyEvent.KEYCODE_I+offset, "\u0630");
    dariKeys.put(KeyEvent.KEYCODE_O, "\u0631");
    dariKeys.put(KeyEvent.KEYCODE_P, "\u0632");
    dariKeys.put(KeyEvent.KEYCODE_A, "\u0698");
    dariKeys.put(KeyEvent.KEYCODE_S, "\u0633");
    dariKeys.put(KeyEvent.KEYCODE_S+offset, "\u0634");
    dariKeys.put(KeyEvent.KEYCODE_D, "\u0635");
    dariKeys.put(KeyEvent.KEYCODE_F, "\u0636");
    dariKeys.put(KeyEvent.KEYCODE_G, "\u0637");
    dariKeys.put(KeyEvent.KEYCODE_G+offset, "\u0638");
    dariKeys.put(KeyEvent.KEYCODE_H, "\u0639");
    dariKeys.put(KeyEvent.KEYCODE_J, "\u063A");
    dariKeys.put(KeyEvent.KEYCODE_K, "\u0641");
    dariKeys.put(KeyEvent.KEYCODE_L, "\u0642");
    dariKeys.put(KeyEvent.KEYCODE_Z, "\u06A9");
    dariKeys.put(KeyEvent.KEYCODE_Z+offset, "\u06AF");
    dariKeys.put(KeyEvent.KEYCODE_X, "\u0644");
    dariKeys.put(KeyEvent.KEYCODE_C, "\u0645");
    dariKeys.put(KeyEvent.KEYCODE_V, "\u0646");
    dariKeys.put(KeyEvent.KEYCODE_B, "\u0648");
    dariKeys.put(KeyEvent.KEYCODE_N, "\u0647");
    dariKeys.put(KeyEvent.KEYCODE_M, "\u06CC");
}

/** This method intercepts hard keys of the phone keyboard to input the Dari letters
@param v The view the key has been dispatched to.
@param keyCode The code for the physical key that was pressed
@param event The KeyEvent object containing full information about the event.
@return True if the listener has consumed the event, false otherwise.
*/

public boolean onKey(View v, int keyCode, KeyEvent event)
{
    if(
        getResources().getConfiguration().hardKeyboardHidden==Configuration.HARDKEYBOARDHIDDEN_NO
        && dariDialogOpen && event.getAction()==KeyEvent.ACTION_DOWN)
    {
        if (keyCode==KeyEvent.KEYCODE_DEL)
            DariHardKeyboardDeleteClick();
        else
        {
            int key=(event.isShiftPressed() || shiftLocked)?keyCode+KeyEvent.getMaxKeyCode():keyCode;
            if (dariKeys.containsKey(key))

```



```

        DariHardKeyboardClick(dariKeys.get(key));
    }
    return true;
}
else
    return false;
}

/** This method sets the appropriate Dari letter to the Dari textbox in the Dari Dialog of the UI.
 * @param key The string to set to the Dari textbox in the Dialog.
 */
public void DariHardKeyboardClick( String key)
{
    String dari = dariDialog.getText();
    if (dari.length() == 0) {
        dariDialog.setText(key);
        if (dari.length() < 2)
            getDariTextView().setText(dari);
        else
        {
            new DariGlyphUtils();
            getDariTextView().setText(DariGlyphUtils.reshapeText(dari).trim());
        }
    }
    else
    {
        dariDialog.setText(dari + key);
        new DariGlyphUtils();
        getDariTextView().setText(DariGlyphUtils.reshapeText(dari+ key).trim());
    }
}

/** This method handles the delete button of the Dari Dialog to delete Dari letters*/
public void DariHardKeyboardDeleteClick()
{
    String s = dariDialog.getText();
    if (s.length() == 0)
        return;
    else if (s.length() == 1)
        dariDialog.setText("");
    else if (s.length() == 2)
        dariDialog.setText(s.substring(0, 1));
    else
        dariDialog.setText(s.substring(0, s.length() - 1));
}

/** This method is onClick() callback for soft Dari Keyboard buttons and handles button press.
 * @param v The view that is selected by the user.
 */
public void DariKeyboardClick(View v)
{
    String key = (String) ((Button) v).getText();
    TextView textView = new TextView(this);
    int orient = getScreenOrientation();
    if (orient == Configuration.ORIENTATION_SQUARE)
    {
        textView.setBackgroundResource(R.drawable.landscape);
        textView.setText(key);
        textView.setTextSize(35);
        Toast toastView = new Toast(this);
        toastView.setView(textView);
    }
}

```

```

        toastView.setDuration	Toast.LENGTH_SHORT);
        int[] loc = new int[2];
        v.getLocationInWindow(loc);
        toastView.setGravity(Gravity.NO_GRAVITY,loc[0]-250,loc[1]-160);
        toastView.show();
    }
    else if (orient == Configuration.ORIENTATION_PORTRAIT)
    {
        textView.setBackgroundResource(R.drawable.portrait);
        textView.setText(key);
        textView.setTextSize(35);
        Toast toastView = new Toast(this);
        toastView.setView(textView);
        toastView.setDuration	Toast.LENGTH_SHORT);
        int[] loc = new int[2];
        v.getLocationInWindow(loc);
        toastView.setGravity(Gravity.NO_GRAVITY,loc[0]-138,loc[1]-77);
        toastView.show();
    }
    String dari = dariDialog.getText();
    if (dari.length() == 0)
    {
        dariDialog.setText(key);
        if (dari.length() < 2)
            getDariTextView().setText(dari);
        else
        {
            new DariGlyphUtils();
            getDariTextView().setText(DariGlyphUtils.reshapeText(dari).trim());
        }
    }
    else
    {
        dariDialog.setText(dari + key);
        new DariGlyphUtils();
        getDariTextView().setText(DariGlyphUtils.reshapeText(dari + key).trim());
    }
}

```

/\*\* This method handles the Delete button press in the Soft Dari Keyboard  
 \* @param v The view selected by the user\*/

```

public void DariKeyboardDeleteClick(View v)
{
    String s = dariDialog.getText();
    if (s.length() == 0)
        return;
    else if (s.length() == 1)
        dariDialog.setText("");
    else if (s.length() == 2)
        dariDialog.setText(s.substring(0, 1));
    else
        dariDialog.setText(s.substring(0, s.length() - 1));
}

```

/\*\* This method handles the Space button press in the Soft Dari Keyboard  
 \* @param v The view selected by the user\*/

```

public void DariKeyboardSpaceClick(View v)
{

```

```

        String s = dariDialog.getText();
        if (s.length() == 0)
            return;
        else
        {
            dariDialog.setText(s + " ");
        }
    }

    /** This method handles the Enter button press in the Soft Dari Keyboard
     * @param v The view selected by the user
     */
    public void DariKeyboardDoneClick(View v)
    {
        String s = dariDialog.getText();
        if(s != null && s.length()>0)
        {
            if ( s.length() == 1)
                getDariTextView().setText(s.trim());
            else
            {
                new DariGlyphUtils();
                getDariTextView().setText(DariGlyphUtils.reshapeText(s).trim());
            }
            String[] translation = getEnglishFromDari(s);
            if (translation == null)
                getEnglishAutoCompleteTextView().setText("Phrase not found");
            else if(translation.length>0)
                getEnglishAutoCompleteTextView().setText(translation[0]);
            dariDialog.setText("");
        }
        dariDialog.dismiss();
        dariDialogOpen = false;
    }

    /** This method handles the Enter button press in the Soft Dari Keyboard
     * @param v The view selected by the user */
    public void DariKeyboardShiftClick(View v)
    {
        if (!shiftLocked)
        {
            shiftLocked=true;
            ((ImageButton)v).setImageResource(R.drawable.sym_keyboard_shift_locked);
            dariDialog.displayExtraletters();
        }
        else
        {
            shiftLocked=false;
            ((ImageButton)v).setImageResource(R.drawable.sym_keyboard_shift);
            dariDialog.displayRegularletters();
        }
    }

    // Callbacks from main.xml
    /** This method displays the soft Dari Keyboard Dialog.
     * @param v The view selected by the user. In this case, it is the Dari Keyboard button in the main window.*/
    public void ShowDariKeyboard(View v)
    {
        inputMethodManager.hideSoftInputFromWindow(getEnglishAutoCompleteTextView().getWindow()

```

```

        owToken(), 0);
        ClearAll(v);
        dariDialogOpen = true;
        showDialog(0);
    }

/** This method forces the android in-built soft English Keyboard to be displayed.
 * @param v The view selected by the user. In this case, it is the English text view.
 */

public void ShowEnglishKeyboard(View v)
{
    //set the autocomplete threshold back to 3 after setting to 100 in
    //DariDialog.DariDialogDBAdapter.OnItemClick to keep the English autocomplete
    //options from popping up after programmatically setting the English translation
    getEnglishAutoCompleteTextView().setThreshold(3);
    ClearAll(v);
    inputMethodManager.toggleSoftInput(InputMethodManager.SHOW_FORCED, 0);
}

/** This method clears all the text from all the Text Views.
 * @param v The view selected by the user. In this case, it is the "Clear All" button in the main window */

public void ClearAll(View v)
{
    getDariTextView().setText("");
    getEnglishAutoCompleteTextView().setText("");
    if (dariDialog != null)
        dariDialog.setText("");
}

// Utility Methods

/** Initializes the views in the main UI. Sets the typeface and style of the text displayed in the appropriate views.
 * Sets the adapter holding auto completion data in the appropriate views. Sets focus of appropriate views.
 * Creates a new instance of AutocompleteDatabaseAdapter.
 */

private void InitializeMembers()
{
    arial = Typeface.createFromAsset(this.getAssets(), "fonts/arial.ttf");
    dariTextView = (TextView) findViewById(R.id.textview_dari);
    databaseAdapter = new AutocompleteDatabaseAdapter(this);
    getDariTextView().setTypeface(getArial());
    getDariTextView().setFocusable(true);
    englishAutoCompleteTextView = (AutoCompleteTextView) findViewById(R.id.autocomplete_english);
    MainFormDBAdapter adapter = new MainFormDBAdapter(databaseAdapter);
    getEnglishAutoCompleteTextView().setAdapter(adapter);
    getEnglishAutoCompleteTextView().setOnItemClickListener(adapter);
    LoadDariKeyMap();
}

/** This checks orientation of the Screen. public so DariDialog can use it also.
 * @return Integer value indicating Screen Orientation
 */

public int getScreenOrientation()
{
    Display getOrient = getWindowManager().getDefaultDisplay();

```

```

int orientation = getOrient.getOrientation();
if(orientation==Configuration.ORIENTATION_UNDEFINED)
{
    Configuration config = getResources().getConfiguration();
    orientation = config.orientation;
    if(orientation==Configuration.ORIENTATION_UNDEFINED)
    {
        //if height and width of screen are equal then it is square orientation
        if(getOrient.getWidth()==getOrient.getHeight())
        {
            orientation = Configuration.ORIENTATION_SQUARE;
        }
        else
        { //if width is less than height than it is portrait
            if(getOrient.getWidth() < getOrient.getHeight())
            {
                orientation =
Configuration.ORIENTATION_PORTRAIT;
            }
            else
            { // if it is not any of the above it will definitely be landscape
                orientation =
Configuration.ORIENTATION_LANDSCAPE;
            }
        }
    }
}
return orientation; // return value is 1 if Portrait and 2 if it is Landscape Mode
}

/** Sets the typeface and style in which the text should be displayed. */
private void SetTypeface()
{
    getEnglishAutoCompleteTextView().setTypeface(getArial());
    getDariTextView().setTypeface(getArial());
    ((TextView) findViewById(R.id.english_button)).setTypeface(getArial());
    ((Button) findViewById(R.id.dari_button)).setTypeface(getArial());
    ((Button) findViewById(R.id.clear_button)).setTypeface(getArial());
}

// Database methods
/** This method queries the database to get all the English medical terms from the database.
 * @return String array of all the English terms contained in the database
 */
public String[] getAllEnglishPhrases()
{
    return databaseAdapter.queryDB("select english from phrase");
}

/** This method queries the database to get all the Dari medical terms from the database.
 * @return String array of all the Dari terms contained in the database
 */
public String[] getAllDariPhrases()
{
    return databaseAdapter.queryDB("select dari from phrase");
}

/** This method queries the database to get the Dari medical term for the given English term
 * @param english English term to be used in the query
 * @return String array containing the result of the query.

```

```

    */
    public String[] getDariFromEnglish(String english)
    {
        return databaseAdapter.queryDB("select dari from phrase where english = " + "'" + english + "'");
    }

    /** This method queries the database to get the English medical term for the given Dari term
     * @param dari Dari term to be used in the query
     * @return String array containing the result of the query.
     */
    public String[] getEnglishFromDari(String dari)
    {
        return databaseAdapter.queryDB("select english from phrase where dari = " + "'" + dari + "'");
    }

    // Overridden methods

    /** This method is created when the Dialog is first created
     * @param i Integer value
     */
    @Override
    public Dialog onCreateDialog(int i)
    {
        if (dariDialog == null)
        {
            dariDialog = new DariDialog(MainForm.this, this);
        }
        return dariDialog;
    }

    @Override
    /** This method is called when the activity is created. Here the activity's UI is inflated and initialization takes place. */
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        inputMethodManager = (InputMethodManager)
this.getSystemService(Context.INPUT_METHOD_SERVICE);
        setContentView(R.layout.main);
        InitializeMembers();
        SetTypeface();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        LayoutInflater inflater=(LayoutInflater)this.getSystemService(LAYOUT_INFLATER_SERVICE);
        View view=inflater.inflate(R.layout.help_layout,(ViewGroup) findViewById(R.id.help_layout));
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setIcon(R.drawable.help);
        builder.setTitle(R.string.app_name);
        builder.setPositiveButton("OK", new DialogInterface.OnClickListener(){public void
onClick(DialogInterface dialog, int id) {dialog.dismiss();}});
        builder.setView(view);
        builder.create();
        builder.show();
        return true;
    }

```

```

/** Private adapter class that links data from a Cursor to a ListView widget.*/

class MainFormDBAdapter extends CursorAdapter implements android.widget.AdapterView.OnItemClickListener
{

    private AutocompleteDatabaseAdapter autocompleteDBAdapter;
    /**
     * Constructor. Sets the reference to autocomplete database adapter.
     */
    public MainFormDBAdapter(AutocompleteDatabaseAdapter helper)
    {
        super(MainForm.this, null);
        autocompleteDBAdapter = helper;
    }
    /** Called by the ListView for the AutoCompleteTextView field to display
     * the text for a particular choice in the list.
     * @param view The TextView used by the ListView to display a particular choice.
     * @param context The context (Activity) to which this form belongs;
     * @param cursor The cursor for the list of choices, positioned to a particular row.
     */
    @Override
    public void bindView(View view, Context context, Cursor cursor)
    {
        final String text = convertToString(cursor);
        ((TextView) view).setTypeface(getArial());
        ((TextView) view).setText(text);
    }

    /** Called by the AutoCompleteTextView field to display the text for a
     * particular choice in the list.
     * @param context The context (Activity) to which this form belongs;
     * @param cursor The cursor for the list of choices, positioned to a particular row.
     * @param parent The ListView that contains the list of choices.
     * @return A new View to hold a particular choice.
     */
    @Override
    public View newView(Context context, Cursor cursor, ViewGroup parent)
    {
        final LayoutInflater inflater = LayoutInflater.from(context);
        final View view = inflater.inflate(android.R.layout.simple_dropdown_item_1line, parent, false);
        return view;
    }

    /** Invoked by the AutoCompleteTextView field to get completions for the current *input.
     * @param constraint The input entered thus far. The resulting query will search for *medical terms that
     * begins with this string.
     * @return A Cursor that is positioned to the first row (if one exists) and managed by the *activity
     */
    @Override
    public Cursor runQueryOnBackgroundThread(CharSequence constraint)
    {
        if (getFilterQueryProvider() != null)
        {
            return getFilterQueryProvider().runQuery(constraint);
        }
        String str = constraint.toString();
        Cursor cursor = autocompleteDBAdapter.getMatchingPhrases(((str != null && str.length() > 0) ?
        str : null), false);
        return cursor;
    }
}

```

```

    }

    /** Called by the AutoCompleteTextView field to get the text that will be
     * entered in the field after a choice has been made.
     * @param Cursor The cursor, positioned to a particular row in the list.
     * @return A String representing the row's text value. (
     */
    @Override
    public String convertToString(Cursor cursor)
    {
        final int columnIndex = cursor.getColumnIndexOrThrow("english");
        final String str = cursor.getString(columnIndex);
        return str;
    }

    /** Called by the AutoCompleteTextView field when a choice has been made
     * by the user.
     * @param listView The ListView containing the choices that were displayed to the user.
     * @param view The field representing the selected choice
     * @param position The position of the choice within the list
     * @param id The id of the row that was chosen
     */
    public void onItemClick(AdapterView<?> listView, View view, int position, long id)
    {
        Cursor cursor = (Cursor) listView.getItemAtPosition(position);
        String dari = cursor.getString(cursor.getColumnIndexOrThrow("dari"));
        getDariTextView().setText(DariGlyphUtils.reshapeText(dari).trim());
    }
}

```



## DariDialog.java

```
package mil.army.arl;
import com.ahmadiv.dari.DariGlyphUtils;
import android.app.Activity;
import android.app.Dialog;
import android.content.Context;
import android.content.res.Configuration;
import android.database.Cursor;
import android.os.Bundle;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.widget.AdapterView;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.CursorAdapter;
import android.widget.LinearLayout;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

/** This class creates the soft Dari keyboard in a Dialog */

public class DariDialog extends Dialog
{
    // fields
    private static Context ctx;
    private static MainForm mainForm;
    private AutoCompleteTextView dariACTV;
    private String dari="";

    /** Constructs a dialog window as part of the specified Activity.
     * @param context The context the Dialog is to run.
     * @param EngDariActivity Reference to the activity. The Dialog is *created and displayed as a part of that activity
     */
    public DariDialog(Context context, MainForm EngDariActivity)
    {
        super(context);
        ctx=context;
        mainForm=EngDariActivity;
    }

    /** This method sets the given string to the Dari textview in the *Dari Keyboard
     * @param dariText String to set to dari textview
     */
    public void setText(String dariText)
    {
        dari=dariText;
        if (dari.length()<2)
            dariACTV.setText(dari);
        else
            dariACTV.setText(DariGlyphUtils.reshapeText(dari).trim());
        dariACTV.requestFocus();
    }

    /** This method returns the value of the private variable that *holds the Dari letters input by *the user.
     * @return Dari String
    */
}
```

```

*/

public String getText()
{
    return dari;
}

@Override
/**This method initializes the Dialog. It checks for the screen *orientation
 * and sets the content of the Dialog based on the orientation.
 */
protected void onCreate(Bundle b)
{
    super.onCreate(b);
    getWindow().requestFeature(Window.FEATURE_NO_TITLE);
    getWindow().getAttributes().gravity=Gravity.BOTTOM;
    int kb = 0;
    int formOrientation = mainForm.getScreenOrientation();
    TableLayout t = null;
    if (formOrientation==Configuration.ORIENTATION_SQUARE)
        t=(TableLayout)LayoutInflater.from(ctx).inflate(R.layout.dari_keyboard, null);
    elseif (formOrientation==Configuration.ORIENTATION_PORTRAIT)
    {
        kb=mainForm.getResources().getConfiguration().hardKeyboardHidden;
        if (kb == 2)
            t=(TableLayout) LayoutInflater.from(ctx).inflate(R.layout.dari_special_keys, null);
        if (kb != 2)
            t=(TableLayout) LayoutInflater.from(ctx).inflate(R.layout.dari_keyboard, null);
    }
    setContentView(t);
    AutocompleteDatabaseAdapter helper=new AutocompleteDatabaseAdapter((Activity)ctx);
    dariACTV = (AutoCompleteTextView) findViewById(R.id.autotextview_dari);
    dariACTV.setTypeface(mainForm.getArial());
    dariACTV.setThreshold(3);
    DariDialogDBAdapter dbAdapter= new DariDialogDBAdapter(helper, DariDialog.this);
    dariACTV.setAdapter(dbAdapter);
    dariACTV.setOnItemClickListener(dbAdapter);
    dariACTV.setOnKeyListener(mainForm);
}

/**
 * This method is used to display the extra letters in the soft Dari Keyboard, when the Shift key is *pressed on the soft
 * Dari keyboard.
 * This method is called when the screen orientation is in landscape mode.
 */
public void displayExtraletters()
{
    TableRow trow1 = (TableRow) findViewById(R.id.r1);
    for(int ctr=0; ctr<trow1.getChildCount(); ctr++)
    {
        LinearLayout ll = (LinearLayout) trow1.getChildAt(ctr);
        Button b = (Button)ll.getChildAt(0);
        b.setText("");
        if(ctr == 1)
            b.setText("\u067E");
        else if(ctr == 4)
            b.setText("\u0686");
        else if(ctr == 7)
            b.setText("\u0630");
        else if(ctr == 9)

```

```

        break;
    }

    TableRow trow2 = (TableRow) findViewById(R.id.r2);
    for(int ctr=0; ctr<trow2.getChildCount(); ctr++)
    {
        LinearLayout ll = (LinearLayout) trow2.getChildAt(ctr);
        Button b = (Button)ll.getChildAt(0);
        b.setText("");
        if(ctr == 1)
            b.setText("\u0634");
        else if(ctr == 4)
            b.setText("\u0638");
        else if(ctr==8)
            break;
    }

    TableRow trow3 = (TableRow) findViewById(R.id.r3);
    for(int ctr=0; ctr<trow3.getChildCount(); ctr++)
    {
        LinearLayout ll = (LinearLayout) trow3.getChildAt(ctr);
        Button b = (Button)ll.getChildAt(0);
        b.setText("");
        if(ctr == 0)
            b.setText("\u06AF");
        else if(ctr==6)
            break;
    }
}

/** This method is called to display Dari letters on the soft Dari Keyboard. */
public void displayRegularletters()
{
    String[] row1={"\u0627","\u0628","\u062A",
        "\u062B","\u062C","\u062D","\u062E","\u062F","\u0631","\u0632"};
    TableRow trow1 = (TableRow) findViewById(R.id.r1);
    for(int ctr=0; ctr<trow1.getChildCount(); ctr++)
    {
        LinearLayout ll = (LinearLayout) trow1.getChildAt(ctr);
        ((Button)ll.getChildAt(0)).setText(row1[ctr]);
        if (ctr==row1.length)
            break;
    }

    String[] row2={"\u0698","\u0633","\u0635","\u0636","\u0637","\u0639","\u063A","\u0641","\u0642"};
    TableRow trow2 = (TableRow) findViewById(R.id.r2);
    for(int ctr=0; ctr<trow2.getChildCount(); ctr++)
    {
        LinearLayout ll = (LinearLayout) trow2.getChildAt(ctr);
        ((Button)ll.getChildAt(0)).setText(row2[ctr]);
        if (ctr==row2.length)
            break;
    }

    String[] row3={"\u06A9","\u0644","\u0645","\u0646","\u0648","\u0647","\u06CC"};
    TableRow trow3 = (TableRow) findViewById(R.id.r3);
    for(int ctr=0; ctr<trow3.getChildCount(); ctr++)
    {
        LinearLayout ll = (LinearLayout) trow3.getChildAt(ctr);
        ((Button)ll.getChildAt(0)).setText(row3[ctr]);
    }
}

```

```

        if (ctr==row3.length)
            break;
    }
}

/** Specializes CursorAdapter to supply choices to a AutoCompleteTextView.
 * Also implements OnItemClickListener to be notified when a choice is made,
 * and uses the choice to update other fields on the Activity form.
 */
class DariDialogDBAdapter extends CursorAdapter
implements android.widget.AdapterView.OnItemClickListener
{
    private AutocompleteDatabaseAdapter autocompleteDBAdapter;
    private DariDialog parent;
    /**
     * Constructor.
     * @param autocompleteDBAdapter The AutocompleteDatabaseAdapter in use by the *outer class object.
     * @param dDialog Reference to the Dialog
     */
    public DariDialogDBAdapter(AutocompleteDatabaseAdapter helper, DariDialog dDialog)
    {
        super(ctx, null);
        autocompleteDBAdapter=helper;
        parent=dDialog;
    }

    /** Called by the ListView for the AutoCompleteTextView field to display
     * the text for a particular choice in the list.
     * @param view The TextView used by the ListView to display a particular choice.
     * @param context The context (Activity) to which this form belongs;
     * @param cursor The cursor for the list of choices, positioned to a particular row.
     */
    @Override
    public void bindView(View view, Context context, Cursor cursor)
    {
        final String text = DariGlyphUtils.reshapeText(convertToString(cursor));
        ((TextView) view).setTypeface(mainForm.getArial());
        ((TextView) view).setGravity(Gravity.RIGHT);
        ((TextView) view).setText(text);
    }

    /** Called by the AutoCompleteTextView field to display the text for a
     * particular choice in the list.
     * @param context The context (Activity) to which this form belongs;
     * @param cursor The cursor for the list of choices, positioned to a particular row.
     * @param parent The ListView that contains the list of choices.
     * @return A new View to hold a particular choice.
     */
    @Override
    public View newView(Context context, Cursor cursor, ViewGroup parent)
    {
        final LayoutInflater inflater = LayoutInflater.from(context);
        final View view = inflater.inflate(
            android.R.layout.simple_dropdown_item_1line, parent, false);
        return view;
    }

    /** Invoked by the AutoCompleteTextView field to get completions for the current *input.
     * @param constraint The input entered thus far. The resulting query will search for
     * medical terms that begins with this string.

```

```

* @return A Cursor that is positioned to the first row (if one exists) and managed by the *activity
*/
@Override
public Cursor runQueryOnBackgroundThread(CharSequence constraint)
{
    if (getFilterQueryProvider() != null)
        return getFilterQueryProvider().runQuery(constraint);
    Cursor cursor = autoCompleteDBAdapter.getMatchingPhrases(((dari != null && dari.length()>0) ?
    dari : null), true);
    return cursor;
}

/** Called by the AutoCompleteTextView field to get the text that will be
 * entered in the field after a choice has been made.
 * @param Cursor The cursor, positioned to a particular row in the list.
 * @return A String representing the row's text value. (
 */
@Override
public String convertToString(Cursor cursor)
{
    final int columnIndex = cursor.getColumnIndexOrThrow("dari");
    final String str = cursor.getString(columnIndex);
    final String dari=DariGlyphUtils.reshapeText(str).trim();
    return dari;
}

/** Called by the AutoCompleteTextView field when a choice has been made
 * by the user.
 * @param listView The ListView containing the choices that were displayed to the user.
 * @param view The field representing the selected choice
 * @param position The position of the choice within the list
 * @param id The id of the row that was chosen
 */
public void onItemClick(AdapterView<?> listView, View view, int position, long id)
{
    Cursor cursor = (Cursor) listView.getItemAtPosition(position);
    String english = cursor.getString(cursor.getColumnIndexOrThrow("english"));
    String dari = cursor.getString(cursor.getColumnIndexOrThrow("dari"));
    // keep the EnglishAutoCompleteTextView dropdown from appearing, set it back to 3 when the
    ENGLISH button is clicked
    mainForm.getEnglishAutoCompleteTextView().setThreshold(100);
    mainForm.getEnglishAutoCompleteTextView().setText(english);
    mainForm.getDariTextView().setText(DariGlyphUtils.reshapeText(dari).trim());
    parent.dismiss();
}
} // End of Class DariDialogDBAdapter
} // End of Class DariDialog

```

## AutocompleteDatabaseAdapter.java

```
package mil.army.arl;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import android.app.Activity;
import android.content.ContentProvider;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.net.Uri;
import android.util.Log;

/** A class to help database access. */

public class AutocompleteDatabaseAdapter extends ContentProvider
{

    private static final String DB_NAME = "MedicalPhrases.db";
    private static final String DB_PATH = "/data/data/mil.army.arl/databases/";
    private static final int DATABASE_VERSION = 1;
    private DatabaseHelper dbHelper;
    private SQLiteDatabase theDatabase;
    private final Activity theActivity;

    /** A helper class to manage database creation and version management. */

    private class DatabaseHelper extends SQLiteOpenHelper
    {
        /**
         * Constructor to create a helper object to create, open or manage database.
         */
        DatabaseHelper(Context context)
        {
            super(context, DB_NAME, null, DATABASE_VERSION);
            createDatabase();
        }

        private boolean dbExists()
        {
            SQLiteDatabase db=null;
            try
            {
                db=SQLiteDatabase.openDatabase(DB_PATH + DB_NAME, null,
                    SQLiteDatabase.OPEN_READONLY);
            }
            catch (SQLiteException e) {}
            if (db!=null)
                db.close();
            return db!=null?true:false;
        }

        public void onCreate(SQLiteDatabase db) {}
    }
}
```

```

public void createDatabase()
{
    if (!dbExists()) {
        this.getReadableDatabase();
        copyDatabase();
    }
}

private void copyDatabase()
{
    try
    {
        InputStream is;
        is = theActivity.getAssets().open("data/"+DB_NAME);
        File f = new File(DB_PATH);
        if (!f.exists())
            f.mkdir();
        OutputStream os=new FileOutputStream(DB_PATH+DB_NAME);
        byte[] buf=new byte[2048];
        int len;
        while ((len=is.read(buf))>0)
            os.write(buf, 0, len);
        os.flush();
        os.close();
        is.close();
    } catch (IOException ex)
    {
        Log.e("DB ERROR", ex.toString());
    }
}

@Override
/** Called when the database needs to be upgraded.
 * @param db The database.
 * @param oldVersion The old database version.
 * @param newVersion The new database version.
 */
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
    File dbFile=new File(DB_PATH+DB_NAME);
    if (dbFile.exists())
        dbFile.delete();
    copyDatabase();
}

}

/**
 * Constructor - Takes the context to allow the database to be
 * opened/created
 * @param activity The Activity that is using the database
 *
 */
public AutocompleteDatabaseAdapter(Activity activity)
{
    this.theActivity = activity;
    dbHelper = this.new DatabaseHelper(activity);
    theDatabase = dbHelper.getWritableDatabase();
}

/** Closes the database. */

```

```

public void close()
{
    dbHelper.close();
}

/** Return a Cursor that returns all words from the database that begins with the given constraint string
 * @param constraint
 *     Specifies the first letters of the words to be listed. If
 *     null, all rows are returned.
 * @return Cursor managed and positioned to the first word, if found
 * @throws SQLException
 *     if query fails
 */
public Cursor getMatchingPhrases(String constraint, boolean dari) throws SQLException
{
    String queryString =
        "SELECT _id, english, dari FROM phrase";

    if (constraint != null)
    {
        // Query for any rows where the state name begins with the
        // string specified in constraint.
        //
        // NOTE:
        // If wildcards are to be used in a rawQuery, they must appear
        // in the query parameters, and not in the query string proper.
        constraint = constraint.trim() + "%";
        queryString += " WHERE " + (dari?"dari":"english") + " LIKE ?";
    }
    String params[] = { constraint };

    if (constraint == null)
    {
        params = null;
    }
    try
    {
        Cursor cursor = theDatabase.rawQuery(queryString, params);
        if (cursor != null) {
            theActivity.startManagingCursor(cursor);
            cursor.moveToFirst();
            return cursor;
        }
    }
    catch (SQLException e)
    {
        Log.e("AutoCompleteDbAdapter", e.toString());
        throw e;
    }
    Log.e("DB MSG", "found no results for" + constraint);
    return null;
}

/** This method queries the database with a given query string.
 * @param query A SQL query string
 * @return String array containing the result of the query
 */
String[] queryDB(String query)
{
    Cursor cursor;
    String[] results = null;

```



```

try
{
    cursor = theDatabase.rawQuery(query, null);
    if(cursor.moveToFirst())
    {
        int ctr = 0;
        results = new String[cursor.getCount()];
        // if there is data available after the cursor's pointer, add
        // it to the ArrayList that will be returned by the method.
        if (!cursor.isAfterLast()) {
            do {
                results[ctr++] = cursor.getString(0);
            } while (cursor.moveToNext());
        }
    }
} catch (SQLException e)
{
    Log.e("DB ERROR", e.toString());
}
return results;
}

```

```

@Override
public int delete(Uri arg0, String arg1, String[] arg2)
{
    // TODO Auto-generated method stub
    return 0;
}

```

```

@Override
public String getType(Uri uri)
{
    // TODO Auto-generated method stub
    return null;
}

```

```

@Override
public Uri insert(Uri uri, ContentValues values)
{
    // TODO Auto-generated method stub
    return null;
}

```

```

@Override
public boolean onCreate()
{
    // TODO Auto-generated method stub
    return false;
}

```

```

@Override
public Cursor query(Uri uri, String[] projection, String selection,
                    String[] selectionArgs, String sortOrder)
{
    // TODO Auto-generated method stub
    return null;
}

```

```

@Override
public int update(Uri uri, ContentValues values, String selection,
                  String[] selectionArgs)

```

```
    {  
        // TODO Auto-generated method stub  
        return 0;  
    }  
}
```

NO. OF COPIES	ORGANIZATION
1 ELEC	ADMNSTR DEFNS TECHL INFO CTR ATTN DTIC OCP 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
7	US ARMY RSRCH LAB ATTN IMNE ALC HRR MAIL & RECORDS MGMT ATTN RDRL CII B R WINKLER ATTN RDRL CII B S METU ATTN RDRL CII T G JAHED ATTN RDRL CII T S LAROCCA ATTN RDRL CIO LL TECHL LIB ATTN RDRL CIO LT TECHL PUB ADELPHI MD 20783-1197

INTENTIONALLY LEFT BLANK.